The Data Capsule for Non-Consumptive Research: Final Report

Beth Plale[1], Indiana University; Atul Prakash, University of Michigan; and Robert McDonald Indiana University

I. Project Goal

The outcome of the HTRC Data Capsule, as per the original proposal, is a prototype system that demonstrates non-consumptive, computational access to a restricted full-text corpus. Expanding upon this, over the last 3 years, the PIs and their students have designed a system that supports researchers and educators located across the country to carry out their own analysis on a restricted corpus using their own tools. The corpus sits in one location and within a Trusted Computing Base (TCB), that is, supported by services that are trusted. The framework we designed in this project gives the corpus guaranteed protection from community research and education use, use that might inadvertently include compromised tools, that is, tools that when run would stream restricted content in whole or in part from the corpus' home in a trusted computing base.

Specifically, the design of the system has been guided by the following requirements:

- *Openness:* Can the framework support users running their own personal analysis tools?
- *Non-consumptive use:* Can the framework be trusted to run personal analysis tools even when they are compromised without compromising protections on the data?
- *Large-scale and low cost:* Can the protections be extended to utilization of large-scale national (public) computational resources for computational access to restricted corpus?

II. Key Outcomes

We built a running system that we call HathiTrust Research Center (HTRC) Data Capsules, which is available as part of HTRC v3.0 released in beta version 16 January 2015. HTRC Data Capsules has a well-defined threat model. That threat model and the system architecture are the most significant technical outcomes of this project. Our outreach and education are substantial non-technical outcomes.

II.A End-User View of the System

The target environment for the system is providing access to restricted datasets where it is desirable to provide access to the datasets to end-users who are generally trusted with remote access to the dataset. End-users can bring in their own software for analysis as well as bring in external datasets. The security risk that the system mitigates is that of software that is used for analysis being inadvertently malicious and thus leaking

[1] For more information contact plale@indiana.edu

data to a third party. Our current prototype permits remote access to the datasets. In situations where datasets are highly sensitive, the proposed technology could be used in more restricted settings where end-users can access the data only from secure rooms, but still have the ability to import their software and external datasets to assist in the analysis over the network from the secure facility, without introducing the risk of leaking restricted datasets.

An example workflow for an end-user wishing to access and analyze the restricted data via the system is as follows:

1. The user goes to a web portal and requests an account on the system. Since our threat model assumes that the user authorized on the system is not malicious, reasonable policies are assumed to be deployed to validate the users (e.g., verifying their organization's email ID (e.g., it is from an appropriate research organization and belongs to the user), and a signed, written agreement that they will not misuse the access to restricted data). Each user is issued a credential for access to the system. A change of email ID should require re-verification of the email ID, since the system relies on a good email ID for securing later steps of this workflow.

2. Via the web portal, the user is assigned a new virtual machine (VM) that will be used to get access to restricted data. The user is issued credentials to access the VM. The user switches the VM between two modes as needed: (1) *secure mode* or (2) *maintenance mode*. In secure mode, the user accesses the restricted data on the trusted network via the VM, but is not granted access to rest of the Internet. In maintenance mode, the user is given full access to the Internet, but is not granted access to the restricted data or to the results from any previous access to restricted data. The user is given administrative access to the VM in both modes, but external access to the network and to secure storage is controlled by trusted software outside the control of the VM and the user.

3. A user will typically first access the VM in maintenance mode, bring in the external datasets, and configure the VM with analysis software (possibly from external sources). When in maintenance mode, the user has full administrative access to the VM. Once the VM is properly configured, the user invokes a switch in the web portal to switch the VM to secure mode.

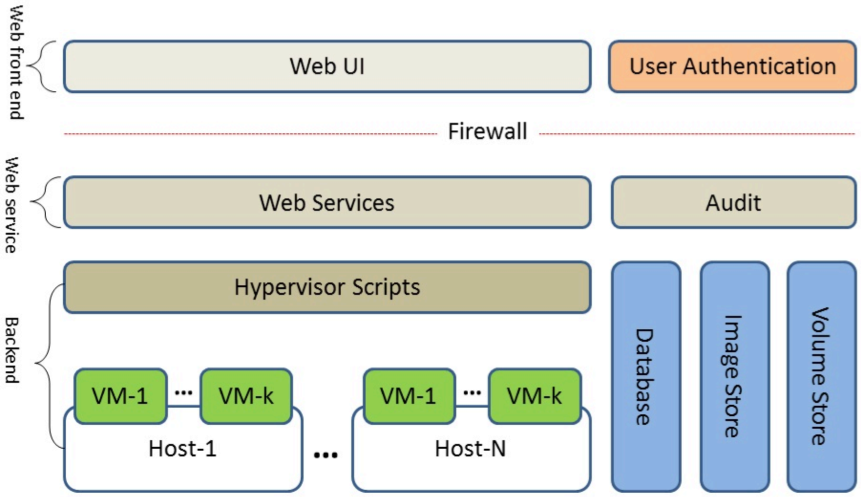
4. Once the VM is in secure mode, the user is able to access the restricted data via the VM and use the software on the VM to analyze the restricted data. Any results from the analysis must be stored on a secure-mode only partition.

5. As needed, the user can switch between maintenance mode (to bring in data and software from external sources) and secure mode (to access the restricted data and the previous results stored on secure storage).

6. The user then requests exporting of the analysis results via the web portal. The produced results are optionally logged and emailed to the user's authorized email ID. We assume that the web portal encrypts the results before sending it over email, with the decryption key provided out-of-band to the user. The reason for emailing the results is that it provides an extra layer of security and an alert to the user in case the user's web portal account is compromised.

II.B Architecture for Secure System

The high-level architecture of the HTRC Data Capsules (see Figure below and discussed in greater length in Zeng et al. 2014) consists of three layers, top to bottom as follows: a web front-end, a web service layer, and a back end. Users of the system use the web front end to create virtual machines, switch them between maintenance and secure modes, and to request releasing of analysis results via email, as described in the previous section. The web services layer translates the web requests from the users and conveys them to the back end that manages the virtual machines. The back end consists of physical machines and hypervisor software that run the virtual machines of the data capsules. It also includes the data capsules implementation that consists of scripts that wrap hypervisor commands to perform several tasks such as restricting the network connectivity of the VMs, attaching or detaching secure storage to them, and switching their state between secure and maintenance modes. The database, image store, and volume store are also part of this layer. The database is used by the web service layer to maintain persistent states for virtual machines and different operations. The image store contains virtual machine images. The volume store contains secure storage volumes that are attached or detached from VMs as needed.



The web service layer is the central controller of the system. It is responsible for resource allocation, request scheduling, state maintenance, and failover on the physical layer. It also has an audit component to log users' activities. To validate user identity, we use OAuth 2.0 for user authentication, though authentication should be further augmented with validation of the user’s identity so that the user can be mapped to a real person or a real organization since we assume that users of the system are benign.

The back end is Data Capsules, a system developed by Borders et al. (2009) that allows privileged access to sensitive data while also restricting the channels through which that data can be released. In this project we extended Capsules to snapshot the VMs at points in time. We added simple policy-based restriction mechanisms to allow a user to enter secure mode. After using the restricted data, when the user returns to normal use of the system (also called maintenance mode), all changes to the system except those made to the attached secure storage are forgotten. In this way, network and storage channels cannot be used to leak the sensitive data. This usage model is readily applicable to non-consumptive use, in that researchers can administer their system (e.g. install any required software tools) and then switch to secure mode to perform analysis in an environment known to be secure.

II. C. Threat Model

The prototype for non-consumptive, computational access to a restricted full-text corpus implements the following threat model:

1. Users access restricted data through remotely accessed VMs that read data from a network-accessed data service.

2. The VM that is given to the user for use is not part of the Trusted Computing Base (TCB). The remaining support is within the TCB: the Virtual Machine Manager (VMM), the host that the VMM runs on, and the system services that enforce network and data access policies for the virtual machines. The HTRC data services themselves are also part of the TCB.

3. We assume the possibility of malware (i.e., malicious software) being installed as well as other remotely initiated attacks on the VM. These attacks could potentially compromise the entire operating system and install a rootkit, both of which are undetectable to the end user.

4. The end users themselves are considered to act in good faith, but this does not preclude the possibility of them unwittingly allowing the system to be compromised. This is a reasonable assumption in case where users are required to sign a use agreement before using the system.

5. Users have VNC access (but not SSH access) to their virtual machines so that they have a graphical interface to the machine. However, this access does admittedly provide a channel for potential data leak. For now, we make the assumption that the end user acts in good faith, and also assume that they are the only one accessing the virtual machine.

   A channel is provided to the user to release and retrieve results when research is complete. The user receives a URL in their email inbox where they can retrieve the results. In the future, released data could be encrypted and undergo automated review to detect potential abuses. Inadvertent release of results via user's email inbox requires that malware compromises exist to a user's account. This is unlikely for users who use their institution emails (which the HathiTrust Research Center requires). Since the user is assumed to be benign in our threat model, users are likely to report an unexpected release of result files from the system.

6. A potential threat is that of covert channels between virtual machines that run on the same host machine. For instance, a virtual machine running in secure mode could possibly make use of such covert channels to leak data to a co-resident virtual machine running in maintenance mode, which can in turn leak the data anywhere it pleases. We currently have a prototype solution to address the

solution – it requires using two physically separated systems, one that only runs VMs in secure mode and another that runs VMs only in maintenance mode.

III. Open Questions

While the project has demonstrated that prototype system can be built for non-consumptive, computational access to a restricted full-text corpus, there remain open questions that expand the usefulness of the system.

Malicious users: End users themselves are considered to act in good faith, but this does not preclude the possibility of them unwittingly allowing the system to be compromised. This is a reasonable assumption in the case where users are required to sign a use agreement before using the system. But feedback from HTRC community has suggested that this assumption needs reexamination.

Distributed computation over "pinned" data collections: Algorithmic and architectural support is needed for computation over two pinned, and distributed data collections, one or both of which contain restricted data.

Big data: The current system limits users to running their analysis on a single VM. While this limitation we think is viable for 80% of the anticipated uses based on experience with HTRC users, the solution needs extending to accommodate analysis that requires a distributed cluster, such as using MapReduce on a university cluster.

Use outside HTRC: The framework of the HTRC Data Capsules has not been applied in settings other than to protect copyrighted content in HTRC. We were part of a small team who convened the CLIR/CNI Workshop on Expanding Access to Research Collections to discuss existing solutions for expanding access to restricted collections. HTRC Data Capsule was discussed frequently as a model, and there was considerable interest.

An interesting extension is to sensitive sensor data. One of our group maintains a repository (Whereabouts) of sensor data from mobile devices and another repository, SenStore, which contains data from sensors on highway bridges in California and Michigan. All these databases contain information that is sensitive in different ways (i.e., copyrighted works, user-identifying data, and security-sensitive data), but there is interest in making them available for research and analytics.

Acknowledgements

This work is funded through a generous grant from the Alfred P. Sloan Foundation, grant # 2011-6-27. Significant project contributions were made by PhD students Alex Cromwell, University of Michigan; Jiaan Zeng and Guangchen Ruan, Indiana University.

References

Borders, Kevin; Weele, Eric Vander; Lau, Billy and Prakash, Atul, Protecting Confidential Data on Personal Computers with Storage Capsules,*18th USENIX Security Symposium (SSYM'09)*, USENIX Association, pp. 367-382, 2009

Jiaan Zeng, Guangchen Ruan, Alexander Crowell, Atul Prakash, Beth Plale 2014. Cloud Computing Data Capsules for Non-consumptive Use of Texts, *5th Workshop on Scientific Cloud Computing, co-located with ACM High Performance Distributed Computing (HPDC),* Vancouver, CA, Jun 2014.